
# SwipeTrack Browser API

*Revision 1.1.2*

# Table of Contents

# Introduction

This document outlines how to integrate a website with the SwipeTrack Browser API. The API outlined in this document allows websites that implement it to communicate with supported devices through the SwipeTrack Browser application.

The SwipeTrack Browser API has two parts to it:

- For websites to receive data from the application, the API assumes the existence of several JavaScript callback functions outlined in the section *SwipeTrack JavaScript Functions*.
- For websites to communicate with and makes requests to the application, the API provides a special URI scheme outlined in the section *SwipeTrack Browser URI Scheme*.

## Release Notes

| Version | Date | Changes |
|---|---|---|
| 1.0.0 | 5/20/2014 | Initial document release |
| 1.0.1 | 6/12/2014 | Added the section on User Defined Callbacks |
| 1.0.2 | 7/15/2014 | Added “stBrowser://printURL?url=@URL” function<br>Added Star Micronics TSP100 to supported devices list<br>Deprecated “stBrowser://airPrint?url=@URL” |
| 1.0.3 | 8/26/2014 | Added methods for capturing a user signature<br>Added “msrRawEncryptionData” to Table 5 |
| 1.1.0 | 12/23/2014 | Added “stBrowserCacheStoragePolicy” to Table 4 |
| 1.1.1 | 1/16/2015 | Added “stBrowserDidCapturePicture” JavaScript callback<br>Added “stBrowser://capturePicture” function |
| 1.1.2 | 1/29/2015 | Updated Supported Devices list<br>Updated “stBrowser://printBarcode” function with new printer support<br>Added section for SwipeTrack Sequence |

# Supported Devices

## Barcode and Magnetic Stripe Readers (MSR)

- Griffin Olli – http://griffintechnology.com/business/olli
- Honeywell Captuvo – http://www.honeywellaidc.com/en-US/Pages/Category.aspx?category=enterprise-sleds-for-apple-devices&cat=HSM
- Infinite Peripherals Linea Pro and Infinea Tab – http://ipcprint.com/
- KoamTac KDC 400 Series – http://koamtac.com/kdc400.html
- SocketMobile – http://www.socketmobile.com/

## Bluetooth Printers

- BlueBamboo PocketPOS P25 – https://bluebamboo.com/products/show.asp?id=16
- Infinite Peripheral DPP Series – http://ipcprint.com/
- Star Micronics Portable Printers – http://www.starmicronics.com/printer/portable_printers

## Network Printers

- Star Micronics Thermal Printers – http://www.starmicronics.com/printer/thermal_printers

## Radio-frequency Identification (RFID) Readers

- Infinite Peripherals Linea Pro and Infinea Tab – http://ipcprint.com/

# SwipeTrack Browser JavaScript Functions

The first part of the SwipeTrack Browser API is the JavaScript interface. Whenever the SwipeTrack Browser application detects certain events, e.g. a barcode scan or a card swipe, it will communicate that information to the client through several callback functions you implement within your website.

## Barcode Reader Callback Functions

```
function stBrowserDidScanBarcode(type, data, stid)
```

***Description:***

Called upon successful scan of a barcode.

***Parameters:***

`type` – integer, represents the type of barcode scanned. Use *API Reference Table 1* for a complete list of recognized barcode types and their values.

`data` – string, the data stored in the scanned barcode.

`stid` – string, SwipeTrack Id.

## Magnetic Stripe Reader (MSR) Callback Functions

```
function stBrowserDidReadCardData(track1, track2, track3, stid)
```

***Description:***

Called upon successful swipe of a magnetic card.

**Note:** This function is only called when a card is swiped using an ***unencrypted*** MSR head.

***Parameters:***

`track1` – string, the data stored in track 1 of the magnetic card.

`track2` – string, the data stored in track 2 of the magnetic card.

`track3` – string, the data stored in track 3 of the magnetic card.

`stid` – string, SwipeTrack Id.

```
function stBrowserDidReadEncryptedCardData(type, data, stid)
```

***Description:***

Called upon successful swipe of a magnetic card.

**Note:** This function is only called when a card is swiped using an ***encrypted*** MSR head.

***Parameters:***

`type` – integer, represents the type of encryption used to encrypt the card data. Use *API Reference Table 2* for a complete list of recognized encryption types and their values.

`data` – object, the encrypted data stored on the magnetic card. Use *API Reference Table 5* for a complete list of possible properties and their associated values. Not all properties may exist; different devices send encrypted data in different ways. Be sure to check that a property exists before attempting to use its value.

`stid` – string, SwipeTrack Id.

```
function stBrowserDidReadFinancialCardData(acct, chName, expMonth, expYear,
fName, lName, stid)
```

***Description:***

Called upon successful swipe of a financial magnetic card. This function is called after *stBrowserDidReadCardData* when a financial magnetic card is swiped.

**Note:** This function is only called when a card is swiped using an ***unencrypted*** MSR head.

***Parameters:***

`acct` – string, account number of swiped financial card.

`chName` – string, card holder name of swiped financial card.

`expMonth` – integer, expiration month of swiped financial card.

`expYear` – integer, expiration year of swiped financial card.

`fName` – string, first name of card holder of swiped financial card.

`lName` – string, last name of card holder of swiped financial card.

`stid` – string, SwipeTrack Id.

## Radio-frequency Identification Reader (RFID) Callback Functions

```
function stBrowserDidDetectRFCard(index, type, serial, stid)
```

***Description:***

Called when a supported RFID card enters the range of the RFID reader.

***Parameters:***

`index` – integer, RF card handle. This value is used for reading and closing the card through the SwipeTrack URI Scheme.

`type` – integer, represents the type of RFID card detected. Use *API Reference Table 3* for a complete list of recognized RFID card types and their values.

`serial` – string, the id of the RFID card that was detected.

`stid` – string, SwipeTrack Id.

```
function stBrowserDidReadRFCard(index, address, length, data, stid)
```

***Description:***

Called after an RF card has been successfully read.

***Parameters:***

`index` – integer, RF card handle.

`address` – integer, address of the read data within the RF card.

`length` – integer, length of the read data.

`data` – string, the data that was read from the RF card, base 64 encoded.

`stid` – string, SwipeTrack Id.

```
function stBrowserShouldCloseRFCardOnDetect()
```

***Description:***

Called by the SwipeTrack Browser application when an RFID card is detected. This function should simply return true or false. If true is returned, the application will immediately close any RFID card that comes into range for further reading. The *stBrowserDidDetectRFIDCard* method will still be called with the cards information. If false is returned the application will leave the RFID card open for further reading, however no new cards can be detected until the open card is closed.

**Note:** ***This function is not required to be implemented.*** If omitted the browser assumes a true value is returned and closes all RFID cards after reading their ID. This function is only required if you need to read data from an RFID card beyond just the ID.

***Parameters:***

No parameters.

## Miscellaneous Callback Functions

```
function stBrowserDidCapturePicture(picture, stid)
```

***Description:***

Called after a user has captured a picture with the device camera.

***Parameters:***

`picture` – string, base 64 representation of the PNG captured image.

`stid` – string, SwipeTrack Id.

```
function stBrowserDidCaptureSignature(signature, stid)
```

***Description:***

Called after a user has provided a signature.

***Parameters:***

`signature` – string, base 64 representation of the PNG signature image.

`stid` – string, SwipeTrack Id.

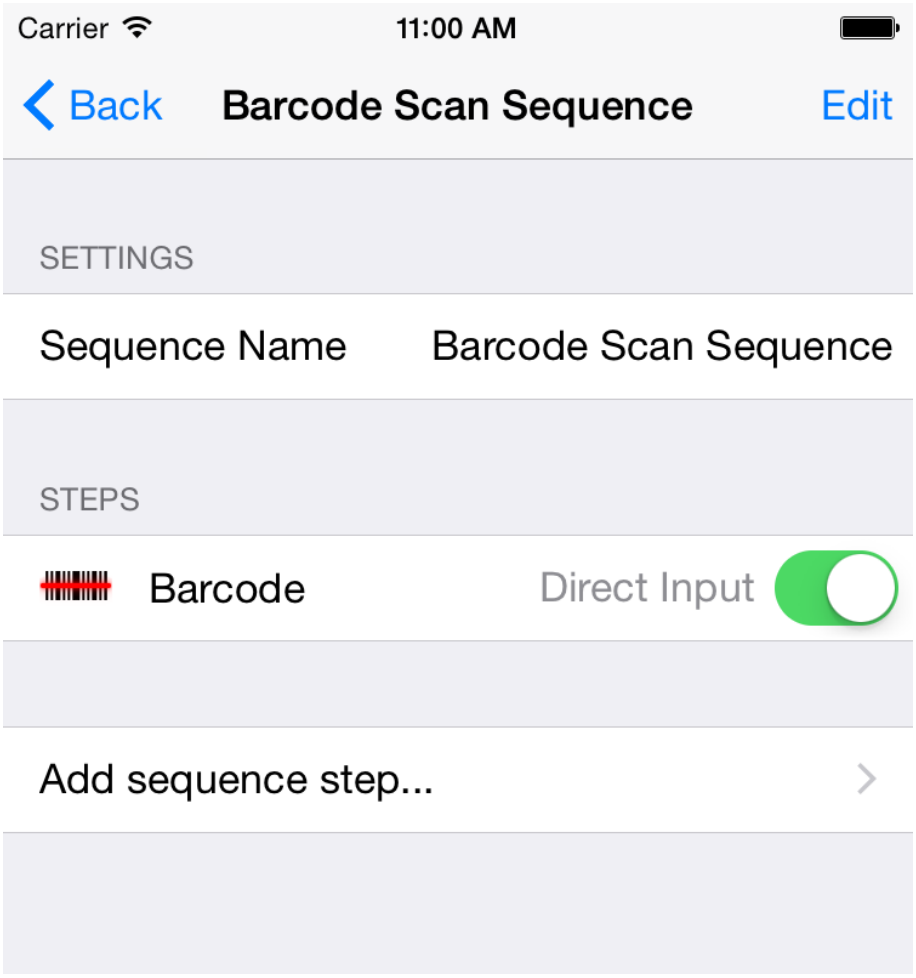
## SwipeTrack Sequence

In addition to the standard callback functions, to allow for a finer grain of control during certain events, is SwipeTrack Sequence. Previously, options like Direct Input would cover both barcode scans and card swipes together not allowing you to control them separately. With SwipeTrack Sequence you can control what happens during barcode scans and card swipes independent of each other.

In addition, SwipeTrack Sequence allows for the injection of JavaScript code both before and after the main sequence event! Simply click “Add sequence step…” to add a JavaScript step, type the your code, and place the step where you want in relation to the main sequence event! SwipeTrack Sequence even includes a built-in JavaScript editor with syntax highlighting!

Note, SwipeTrack Sequence runs after all other API callbacks are made.

To access SwipeTrack Sequence, open the settings window of the SwipeTrack Browser application by triple tapping on your iOS device and navigating to the “Browser Settings” section. In there you will find the “SwipeTrack Sequence” section.



# SwipeTrack Browser URI Scheme

The second part of the SwipeTrack Browser API is the SwipeTrack Browser URI interface. In order for websites to communicate with and make requests to the SwipeTrack Browser application a special URI, *stBrowser://,* is provided. This URI is used by formatting a URI string in accordance with one of the supported functions below then setting the JavaScript *window.location* object to point to the URI.

## Barcode Functions

`stBrowser://startBarcodeScanner`

***Description:***

Immediately attempts to start the barcode scanner.

***Parameters:***

No parameters.

`stBrowser://startCameraScanner`

***Description:***

Immediately attempts to start the camera barcode scanner.

***Parameters:***

No parameters.

`stBrowser://stopBarcodeScanner`

***Description:***

Immediately attempts to stop the barcode scanner.

***Parameters:***

No parameters.

`stBrowser://stopCameraScanner`

***Description:***

Immediately attempts to stop the camera barcode scanner.

***Parameters:***

No parameters.

## Printing Functions

`stBrowser://printBarcode?data=@DATA&type=@TYPE`

***Description:***

Prints to the connected printer a barcode with the given data as the specified type.

**Note:** This method works with the BlueBamboo P25 series printer and Infinite Peripherals DPP series printers.

BlueBamboo P25 Supported Barcodes

| Value | Name |
|---|---|
| 7 | BARCODE_CODE_128 |
| 12 | BARCODE_EAN_8 |
| 13 | BARCODE_EAN_13 |
| 31 | BARCODE_UPC_A |
| 35 | BARCODE_UPC_E |
| 1009 | BARCODE_PDF417 |

Infinite Peripherals DPP Supported Barcodes

| Value | Name |
|---|---|
| 1 | BARCODE_CODABAR |
| 5 | BARCODE_CODE_39 |
| 6 | BARCODE_CODE_93 |
| 7 | BARCODE_CODE_128 |
| 12 | BARCODE_EAN_8 |
| 13 | BARCODE_EAN_13 |
| 18 | BARCODE_ITF_14 |
| 31 | BARCODE_UPC_A |
| 35 | BARCODE_UPC_E |
| 1009 | BARCODE_PDF417 |
| 1011 | BARCODE_QRCODE |

***Parameters:***

**`@DATA`** – string, the data to encode into the barcode.

**`@TYPE`** – integer, type of barcode to encode the data with. Use *API Reference Table 1* for a complete list of barcode types and their values.

`stBrowser://printHtml?html=@HTML`

***Description:***

Renders and prints to the connected printer a given html page. CSS styling is supported.

**Note:** The BlueBamboo P25 series printer supports a maximum page width of 384 pixels.

***Parameters:***

**`@HTML`** – string, the raw html (including CSS styling) to render and print.

`stBrowser://printText?text=@TEXT`

***Description:***

Prints to the connected printer a text string.

***Parameters:***

**`@TEXT`** – string, the text to print.

`stBrowser://printURL?url=@URL`

***Description:***

Renders and prints to the connected printer a given URL. CSS styling is supported.

***Parameters:***

**`@URL`** – string, the URL to render and print.

## RFID Functions

`stBrowser://closeRFCard?index=@INDEX`

***Description:***

Closes the RFID card located at the provided index for further reading.

***Parameters:***

**`@INDEX`** – integer, RF card handle.

`stBrowser://readRFCard?index=@INDEX&address=@ADDRESS&length=@LENGTH`

***Description:***

Attempts to read data stored on an RFID card at the requested address.

**Note:** Make sure any required RFID card keys are set prior to calling method.

***Parameters:***

**`@INDEX`** – integer, RF card handle.

**`@ADDRESS`** – integer, address of the data to read within the RF card.

**`@LENGTH`** – integer, length of data to read.

## SwipeTrack Browser Functions

`stBrowser://capturePicture`

***Description:***

Presents the user with an image capture screen. Once the user has captured a picture, the *stBrowserDidCapturePicture* function will be called.

***Parameters:***

No parameters.

`stBrowser://captureSignature?field=@FIELD`

***Description:***

Presents the user with a signature dialog box. Once the user has finished their signature, the *stBrowserDidCaptureSignature* function will be called.

***Parameters:***

**`@FIELD`** – string, optional, name of the form field whose "value" attribute will be populated with a base 64 representation of the PNG signature data.

`stBrowser://clearCredentials`

***Description:***

Removes any stored credentials for the host in the browser home page setting.

***Parameters:***

No parameters.

`stBrowser://getAppSettings?callback=@CALLBACK`

***Description:***

Gets the SwipeTrack Browser application settings in the form of a JSON dictionary. Use *API Reference Table 4* for a complete list of dictionary keys and a description of the contained values.

**Note:** Not all supported sled devices support all of the available sled device settings. Only the supported sled device settings of the connected sled device will be returned via this method.

***Parameters:***

**`@CALLBACK`** – string, name of the callback function that will be called and passed the settings dictionary.

**Note:** The callback function must accept two parameters; the first parameter will be the settings dictionary and the second parameter will be the SwipeTrack Id.

`stBrowser://playSound?sound=@SOUND`

***Description:***

Plays a sound through the attached iOS device.

**Note:** The following sound names are available for play:

| **Sound Name** |
|---|
| attention |
| error |
| success |

***Parameters:***

**`@SOUND`** – string, the name of the sound to play.

`stBrowser://quitApplication?msg=@MSG`

***Description:***

Quits the SwipeTrack Browser application, displaying a closing message.

***Parameters:***

**`@MSG`** – string, optional, the message to display to the user before quitting the application.

`stBrowser://setAddressBarHidden?hidden=@VALUE`

***Description:***

Sets the visibility of the address bar in the SwipeTrack Browser application.

***Parameters:***

**`@VALUE`** – string, recognized values are “TRUE” and “FALSE”.

`stBrowser://setAppSettings?settings=@SETTINGS`

***Description:***

Sets the SwipeTrack Browser application settings from the JSON dictionary. Use *API Reference Table 4* for a complete list of dictionary keys and a description of the contained values.

***Parameters:***

**`@SETTINGS`** – string, JSON representation of the settings dictionary containing the keys and values of the settings to set.

`stBrowser://setHomePage?page=@PAGE`

***Description:***

Sets the home page setting of the SwipeTrack Browser application.

***Parameters:***

**`@PAGE`** – string, the home page address to set.

`stBrowser://setNavigationBarHidden?hidden=@VALUE`

***Description:***

Sets the visibility of the navigation bar in the SwipeTrack Browser application.

***Parameters:***

**`@VALUE`** – string, recognized values are “TRUE” and “FALSE”.

`stBrowser://vibrateDevice?count=@COUNT`

***Description:***

Vibrates, if supported, the iOS device the SwipeTrack Browser application is installed on.

***Parameters:***

**`@COUNT`** – integer, number of times to vibrate the device.

# API Reference

## Table 1 – STBarcodeType Enumeration Values

| Value | Name |
|---|---|
| 1 | BARCODE_CODABAR |
| 2 | BARCODE_CODE_11 |
| 3 | BARCODE_CODE_25_ITF |
| 4 | BARCODE_CODE_25_NITF |
| 5 | BARCODE_CODE_39 |
| 6 | BARCODE_CODE_93 |
| 7 | BARCODE_CODE_128 |
| 8 | BARCODE_CPC_BINARY |
| 9 | BARCODE_DUN_14 |
| 10 | BARCODE_EAN_2 |
| 11 | BARCODE_EAN_5 |
| 12 | BARCODE_EAN_8 |
| 13 | BARCODE_EAN_13 |
| 14 | BARCODE_FIM |
| 15 | BARCODE_GS1_128 |
| 16 | BARCODE_GS1_DATABAR |
| 17 | BARCODE_INTELLIGENT_MAIL |
| 18 | BARCODE_ITF_14 |
| 19 | BARCODE_KIX |
| 20 | BARCODE_MODIFIED_PLESSEY |
| 21 | BARCODE_MTF |
| 22 | BARCODE_PHARMACODE |
| 23 | BARCODE_PLANET |
| 24 | BARCODE_PLESSEY |
| 25 | BARCODE_POSTBAR |
| 26 | BARCODE_POSTNET |
| 27 | BARCODE_RM4SCC |
| 28 | BARCODE_TELEPEN |
| 29 | BARCODE_UPC_2 |
| 30 | BARCODE_UPC_5 |
| 31 | BARCODE_UPC_A |
| 32 | BARCODE_UPC_B |
| 33 | BARCODE_UPC_C |
| 34 | BARCODE_UPC_D |
| 35 | BARCODE_UPC_E |
| 1000 | BARCODE_AZTEC |
| 1001 | BARCODE_CYBERCODE |
| 1002 | BARCODE_DATAGLYPH |
| 1003 | BARCODE_DATAMATRIX |
| 1004 | BARCODE_DATASTRIP |
| 1005 | BARCODE_EZCODE |
| 1006 | BARCODE_MAXICODE |
| 1007 | BARCODE_MICRO_PDF417 |

| | |
|---|---|
| 1008 | BARCODE_MICRO_QRCODE |
| 1009 | BARCODE_PDF417 |
| 1010 | BARCODE_QODE |
| 1011 | BARCODE_QRCODE |
| 1012 | BARCODE_SHOTCODE |
| 1013 | BARCODE_SPARQCODE |
| 9999 | BARCODE_UNKNOWN |

### Table 2 – STMSREncryptionType Enumeration Values

| Value | Name |
|---|---|
| 1 | ALGORITHM_3DES |
| 2 | ALGORITHM_AES128 |
| 3 | ALGORITHM_AES256 |
| 4 | ALGORITHM_ECC |
| 5 | ALGORITHM_IDTECH |
| 6 | ALGORITHM_MAGTEK |
| 7 | ALGORITHM_RSA |
| 9999 | ALGORITHM_UNKNOWN |

### Table 3 – STRFCardType Enumeration Values

| Value | Name |
|---|---|
| 1 | RFCARD_MIFARE_CLASSIC_1K |
| 2 | RFCARD_MIFARE_CLASSIC_4K |
| 3 | RFCARD_MIFARE_DESFIRE |
| 4 | RFCARD_MIFARE_MINI |
| 5 | RFCARD_MIFARE_PLUS |
| 6 | RFCARD_MIFARE_ULTRALIGHT |
| 7 | RFCARD_MIFARE_ULTRALIGHT_C |
| 9999 | RFCARD_UNKNOWN |

### Table 4 – SwipeTrack Browser Application Settings Dictionary Keys

| Key | Value Description |
|---|---|
| iOSDisableDeviceSleep | String indicating whether the automatic iOS device sleep is disabled while the SwipeTrack Browser application is running. Possible values are “TRUE” and “FALSE”. |
| iOSEnableCameraScanning | String indicating whether or not camera scanning is enabled. Possible values are “TRUE” and “FALSE” |
| sledDeviceBarcodeScanMode | String indicating the barcode scanning mode. Possible values are “MOTION”, “MULTI”, and “SINGLE”. |
| sledDeviceBarcodeScannerTimeout | Integer indicating the number of seconds the barcode reader will remain on before timing out. |

| | |
|---|---|
| sledDeviceDisconnectTimeout | Integer indicating the number of inactive seconds the sled device will remain connected before timing out. |
| sledDeviceEnableBattery | String indicating whether or not the connected iOS device can be charged using the sled devices battery. Possible values are “TRUE” and “FALSE”. |
| sledDeviceEnableDevice | String indicating whether or not the connected sled device is enabled. Possible values are “TRUE” and “FALSE”. |
| sledDeviceEnableFastCharging | String indicating whether or not fast charging is enabled. Possible values are “TRUE” and “FALSE”. |
| sledDeviceEnablePassThroughSync | String indicating whether or not pass through sync is enabled. Possible values are “TRUE” and “FALSE”. |
| sledDeviceEnableRFID | String indicating whether or not RFID cards will be detected. Possible values are “TRUE” and “FALSE”. |
| sledDeviceEnableSound | String indicating whether or not the connected sled device will make a scan “beep” sound. Possible values are “TRUE” and “FALSE”. |
| sledDeviceIdleTimeout | Integer indicating the number of inactive seconds the sled device will remain active before timing out and entering an idle state. |
| sledDeviceMIFAREKey | String containing the MIFARE RFID key. |
| sledDeviceMIFAREUltralightKey | String containing the MIFARE Ultralight RFID key. |
| socketMobileEnableDevice | String indicating whether or not SocketMobile scanners are enabled. Possible values are “TRUE” and “FALSE”. |
| stBrowserAddressBarHidden | String indicating the visibility of the address bar. Possible values are “TRUE” and “FALSE”. |
| stBrowserCacheStoragePolicy | String indicating the cache storage policy of the SwipeTrack browser. Possible values are “ALLOWED”, “MEMORY_ONLY”, and “NOT_ALLOWED”. |
| stBrowserCredentialPersistence | String containing the credential persistence value. Possible values are “NEVER”, “RUNNING”, and “FOREVER”. |
| stBrowserDirectInputModeDelimiter | String containing the direct input mode delimiter. |
| stBrowserEnableDirectInputMode | String indicating whether or not direct input mode is enabled. Possible values are “TRUE” and “FALSE”. |
| stBrowserHomePage | String containing the home page URL. |
| stBrowserNavigationBarHidden | String indicating the visibility of the navigation bar. Possible values are “TRUE” and “FALSE”. |
| stBrowserResetOnResume | String indicating whether or not the application should reload the home page and remove any saved credentials when the application loads or resumes. Possible values are “TRUE” and “FALSE”. |

### Table 5 – Encrypted Card Data Object Properties

| **Property Name** | **Property Description** |
|---|---|
| msrEncryptedData | Will only contain raw encrypted card data. |
| msrRawEncryptionData | Will only contain the full, unmodified encryption package from the device. |
| track1EncryptedData | Will only contain track 1, encrypted. |
| track2EncryptedData | Will only contain track 2, encrypted. |
| track3EncryptedData | Will only contain track 3, encrypted. |
| track1MaskedData | Will only contain a masked representation of track 1. |
| track2MaskedData | Will only contain a masked representation of track 2. |
| track3MaskedData | Will only contain a masked representation of track 3. |